Published before October 2000
Terms used **WOLF legacy**

Found **86** of **110,136**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 86 Result page: **1** 2 3 4 5 next

Relevance scale

1 Saving legacy with objects
W. C. Dietrich, L. R. Nackman, F. Gracer
September 1989 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 24 Issue 10
Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

Developers of application software must often work with "legacy systems." These are systems that have evolved over many years and are considered irreplaceable, either because it is thought that duplicating their function would be too expensive, or because they are trusted by users. Because of their age, such systems are likely to have been implemented in a conventional language with limited use of data abstraction or encapsulation. The lack of abstraction complicates adding new ...

2 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

3 A C++ data model supporting reachability analysis and dead code detection
Yih-Farn R. Chen, Emden R. Gansner, Eleftherios Koutsofios
November 1997 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th European conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 22 Issue 6
Full text available: pdf(1.33 MB) Additional Information: full citation, references, citings, index terms

4 Information technology at the turn of teh millennium: past, present, and future trends

Tor J. Larsen, Linda Levine
March 1999 **ACM SIGMIS Database**, Volume 30 Issue 2
Full text available: pdf(397.87 KB) Additional Information: full citation

5 Software engineering for security: a roadmap
Premkumar T. Devanbu, Stuart Stubblebine
May 2000 **Proceedings of the Conference on The Future of Software Engineering**
Full text available: pdf(1.71 MB) Additional Information: full citation, references, citings, index terms

Keywords: copy protection, security, software engineering, water-marking

6 Databases and datastructuring: An object-oriented persistent database interface for CAD
M. N. Sim, P. M. Dewilde
March 1990 **Proceedings of the conference on European design automation**
Full text available: pdf(506.86 KB) Additional Information: full citation, abstract, references, citings

Much activity in the database community is focused on providing database support to the application programmer in a user-friendly environment that improves productivity and encourages software re-use. Object-oriented and Persistent programming are two emerging paradigms that are considered essential to create such an environment. One approach is to converge programming languages and databases to a coherent and consistent whole. Such systems are not yet ready to replace database systems with proc ...

7 Integrating object-oriented programming and protected objects in Ada 95
A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell
May 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 3
Full text available: pdf(245.47 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is a now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming langua ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

8 Documentation: Documenting-in-the-large vs. documenting-in-the-small
Scott R. Tilley
October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing - Volume 2**
Full text available: pdf(646.16 KB) Additional Information: full citation, abstract, references, citings

There is a significant difference between documenting large programs and documenting small ones. By large programs we mean on the order of 1,000,000 lines, usually written by many different people over a long period of time. Most software documentation may be termed *documenting-in-the-small*, since it typically describes the program at the algorithm

and dat a structure level. To understand large legacy systems, one needs *documenting-in-the-large*: documentation describing the high-le ...

Keywords: documenting-in-the-large, reverse engineering, software evolution

9 Precise interprocedural chopping
Thomas Reps, Genevieve Rosay
October 1995 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 3rd ACM SIGSOFT symposium on Foundations of software engineering**, Volume 20 Issue 4
Full text available: pdf(1.29 MB) Additional Information: full citation, references, citings, index terms

Keywords: debugging, graph reachability, interprocedural analysis, program chopping, program dependence graph, program slicing, realizable path

10 Evolution of the conversation machine: a case study of bringing advanced technology to the marketplace
Catherine G. Wolf, Wlodek Zadrozny
January 1998 **Proceedings of the SIGCHI conference on Human factors in computing systems**
Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: business transformation, design rationale, natural language, requirements, speech recognition

11 Recovering software architecture from multiple source code analyses
Melissa P. Chase, Steven M. Christey, David R. Harris, Alexander S. Yeh
July 1998 **ACM SIGPLAN Notices , Proceedings of the 1998 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering**, Volume 33 Issue 7
Full text available: pdf(991.04 KB) Additional Information: full citation, abstract, references, index terms

We describe the experiences we have had in using ManSART - a software architecture recovery tool that we developed and are employing in the analysis of large scale legacy software systems. ManSART uses a battery of standard data flow, control flow, and program slicing capabilities to automatically recover architectural features from source code. This source code analysis is enabled by representations called analysis. Analysis modules describe the interfaces of each component in a multiple compon ...

12 Software evolution: Understanding software systems using reverse engineering technology perspectives from the Rigi project
Hausi A. Müller, Scott R. Tilley, Kenny Wong
October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1**
Full text available: pdf(785.90 KB) Additional Information: full citation, abstract, references

Software engineering research has focused mainly on software construction and has neglected software maintenance and evolution. Proposed is a shift in research from synthesis to analysis. Reverse engineering is introduced as a possible solution to program understanding and software analysis. Presented is reverse engineering technology

developed as part of the Rigi project. The Rigi approach involves the identification of software artifacts in the subject system and the aggregation of these artif ...

Keywords: legacy software, program understanding, reverse engineering, software evolution

13 Linux as a case study: its extracted software architecture
Ivan T. Bowman, Richard C. Holt, Neil V. Brewster
May 1999 **Proceedings of the 21st international conference on Software engineering**
Full text available: pdf(1.00 MB) Additional Information: full citation, references, citings, index terms

Keywords: architecture recovery, redocumentation, software architecture

14 Summary of the Dagstuhl workshop on software architecture
David Garlan, Walter Tichy, Frances Paulisch
July 1995 **ACM SIGSOFT Software Engineering Notes**, Volume 20 Issue 3
Full text available: pdf(1.96 MB) Additional Information: full citation, citings, index terms

15 Principled design of the modern Web architecture
Roy T. Fielding, Richard N. Taylor
June 2000 **Proceedings of the 22nd international conference on Software engineering**
Full text available: pdf(217.34 KB) Additional Information: full citation, abstract, references, citings, index terms

The World Wide Web has succeeded in large part because its software architecture has been designed to meet the needs of an Internet-scale distributed hypermedia system. The modern Web architecture emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems. In this paper, we introduce the Representational State Tra ...

Keywords: WWW, software architectural style, software architecture

16 CHIME: customizable hyperlink insertion and maintenance engine for software engineering environments
P. Devanbu, Y.-F. Chen, E. Gansner, H. Müller, J. Martin
May 1999 **Proceedings of the 21st international conference on Software engineering**
Full text available: pdf(1.28 MB) Additional Information: full citation, references, citings, index terms

17 A Mathematical Program Generator MPGENR
William M. Michaels, Richard P. O'Neill
March 1980 **ACM Transactions on Mathematical Software (TOMS)**, Volume 6 Issue 1
Full text available: pdf(754.45 KB) Additional Information: full citation, references, citings, index terms

18 Workshop on compositional software architectures: workshop report
May 1998 **ACM SIGSOFT Software Engineering Notes**, Volume 23 Issue 3
Full text available: pdf(2.91 MB) Additional Information: full citation, index terms

19 System-level power optimization: techniques and tools
Luca Benini, Giovanni de Micheli
April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2
Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, index terms

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic sytems consisting of a hardware platform and software layers. We consider the three major constituents of hardware that consume energy, namely computation, communication, and storage units, and we review methods of reducing their energy consumption. We also study models for analyzing the energy cost of software, and methods for energy-efficient software design and compilation. This survery ...

20 An architecture for WWW-based hypercode environments
Gail E. Kaiser, Stephen E. Dossick, Wenyu Jiang, Jack Jingshuang Yang
May 1997 **Proceedings of the 19th international conference on Software engineering**
Full text available: pdf(1.84 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 86 Result page: **1** 2 3 4 5 next